

《C/C++ 学习指南》

第18.2讲：头文件及include的用法

作者：邵发 QQ群：417024631
官网：<http://www.afanihao.cn/>
答疑：<http://www.afanihao.cn/kbase/>

C/C++学习指南 邵发 www.afanihao.cn

引例

```
////////// other.cpp //////////  
#include <stdio.h>  
struct Student  
{  
    int id;  
    char name[32];  
};  
void test(Student* s)  
{  
    printf("Student ID: %d , Name: %s \n",  
        s->id, s->name);  
}
```

记住：每个cpp的编译就互相独立的，互相没有任何影响（compile阶段）

引例

```
////////// main.cpp //////////  
#include <stdio.h>  
// 自定义类型  
struct Student  
{  
    int id;  
    char name[32];  
};  
// 函数声明  
void test(Student* s);  
int main()  
{  
    Student s = {1234, "shaofa"};  
    test(&s); // 调用函数  
    return 0;  
}
```

引例

这些写法存在以下问题

- (1) 同一个类型，Student，被重复写了多次
- (2) 扩展性差：如果向Student中再添加一个成员，需要把每一处类型定义都修改一次。
- (3) 定义可能不一致，编译器无法检查。

记住：每个cpp的编译就互相独立的，互相没有任何影响（compile阶段）

使用头文件

头文件：Header File ，通常后缀名为 .h

头文件的写法：

- (1) 后缀名一般为 .h
- (2) 内容一般为几种：类型定义、函数声明、变量声明。

用头文件解决问题：例如，添加一个Student.h，

```
struct Student
{
    int id;
    char name[32];
};
```

使用头文件

头文件：Header File ，通常后缀名为 .h

头文件的使用：

```
#include "xxx.h"
```

使用#include指令，加上文件名，文件名以双引号括起来

#include 指令的原理

#include 的原理：

编译器在处理每个cpp之前，首先进行一个**预处理**：将所有的#include行替换成头文件的具体内容，形成一个中间文件，然后再对这个中间文件进行编译。

这个在编译之前的预处理过程，称为“预编译”过程。

注：如果头文件里还有#include，则反复替换，直到没有任何#include指令为止。

#include 指令的原理

源文件

```
#include "Student.h"

int main()
{
    ...
}
```



预处理之后的中间文件

```
struct Student
{
    int id;
    char name[32];
};

int main()
{
    ...
}
```

#include 指令的原理

所以，一个源文件有没有语法错误，要在预处理之后才能决定。

例1：下面这种情形有没有语法错误？

```
#include "Student.h"
#include "Student.h" // 重复包含了2次
int main()
{
    ...
}
```

#include 指令的原理

例2：

```
/// abc.h ///
int test(int a);

/// main.cpp ///
#include "abc.h"
int main()
{
    test();
    return 0;
}
```

头文件的内容

(1) 类型定义

如struct类型的定义

```
struct Student
{
    int id;
    char name[32];
};
```

头文件的内容

(2) extern声明

函数声明和全局变量的声明

```
extern void test(); // 函数声明，其中extern可以省
extern int number; // 全局变量的声明
```